CLAIMS

1. (Currently amended) A disk controller for implementing efficient disk I/O for a computer system, comprising:

a bus interface for interfacing with a processor and a system memory of the computer system;

a disk I/O engine coupled to the bus interface; and

a device interface coupled to the disk I/O engine for interfacing the disk I/O engine with a disk drive, wherein the disk I/O engine is configured to cause a start up of the disk drive upon receiving a disk start up command from the processor, the start up command configured to hide a start latency of the disk drive, the disk I/O engine further configured to execute a disk transaction by processing the disk transaction information from a memory mapped bypass register coupled to the disk I/O engine.

2. (Withdrawn) The disk controller of claim 1, wherein the bus interface is configured to interface with the processor and the system memory of the computer system in accordance with a hyper transport protocol.

3. (Withdrawn) The disk controller of claim 1, wherein the device interface is configured to coupled to a serial ATA interface of the disk drive.

4. (Withdrawn) The disk controller of claim 1, wherein the device interface is configured to couple to an IDE interface of the disk drive.

5. (Withdrawn) The disk controller of claim 1, further comprising:

a completion status register coupled to the disk I/O engine configured to notify the disk I/O engine and indicate a completion of a pending disk I/O command.

6. (Original) The disk controller of claim 1, further comprising:

a CPB pointer buffer coupled to the disk I/O engine for dynamically appending a plurality of CPB pointers to extend to a number of disk transactions scheduled for execution by the disk I/O engine.

7. (Original) The disk controller of claim 1, further comprising:

a chain memory coupled to the disk I/O engine for buffering a plurality of CPBs to extend to a number of disk transactions scheduled for execution by the disk I/O engine.

8. (Currently amended) A bridge component for implementing efficient disk I/O for a computer system, comprising:

a bus interface for interfacing with a processor and a system memory of the computer system;

a disk controller for executing disk I/O transactions for the computer system, the disk controller further comprising:

a disk I/O engine coupled to the bus interface; and

a device interface coupled to the disk I/O engine for interfacing the disk I/O engine with a disk drive, wherein the disk I/O engine is configured to cause a start up of the disk drive upon receiving a disk start up command from the processor, the disk start up command configured to hide a start latency of the disk drive, the disk I/O engine further configured to execute a disk transaction by processing the disk transaction information from a memory mapped bypass register coupled to the disk I/O engine.

9. (Withdrawn) The bridge component of claim 8, wherein the bridge component includes a plurality of disk controllers for implementing a plurality of channels for a corresponding plurality of disk drives.

10. (Withdrawn) The bridge component of claim 9, wherein at least one of the channels is a serial ATA channel.

11. (Withdrawn) The disk controller of claim 8, further comprising:

a completion status register coupled to the disk I/O engine configured to notify the disk I/O engine and indicate a completion of a pending disk I/O command.

12. (Original) The disk controller of claim 8, further comprising:

a CPB pointer buffer coupled to the disk I/O engine for dynamically appending a plurality of CPB pointers to extend to a number of disk transactions scheduled for execution by the disk I/O engine.

13. (Original) The disk controller of claim 8, further comprising:

a chain memory coupled to the disk I/O engine for buffering a plurality of CPBs to extend to a number of disk transactions scheduled for execution by the disk I/O engine.

14. (Withdrawn) A computer system configured to implement efficient disk I/O, comprising:

a processor;

a system memory coupled to the processor;

a bridge component coupled to the processor; and

a disk controller coupled to the bridge component, the disk controller including a plurality of bypass registers, wherein the processor executes software code stored in the system memory, the software code causing the computer system to implement a method comprising:

transferring a command from the processor to the disk controller, the command causing a start up of a disk drive coupled to the disk controller;

preparing disk transaction information by packaging a plurality of data structures comprising the disk transaction;

transferring the disk transaction information to the bypass registers of the disk controller;

implementing a disk I/O, wherein the disk controller processes the disk transaction information to control the disk drive.

15. (Withdrawn) The computer system of claim 14, wherein the bridge component includes a plurality of disk controllers for implementing a plurality of channels for a corresponding plurality of disk drives.

16. (Withdrawn) The computer system of claim 15, wherein at least one of the channels is a serial ATA channel.

17. (Withdrawn) The computer system of claim 16, further comprising:

a completion status register coupled to the disk I/O engine configured to notify the disk I/O engine and indicate a completion of a pending disk I/O command.

18. (Withdrawn) The computer system of claim 17, further comprising:

a CPB pointer buffer coupled to the disk I/O engine for dynamically appending a plurality of CPB pointers to extend to a number of disk transactions scheduled for execution by the disk I/O engine.

19. (Withdrawn) The computer system of claim 18, further comprising:

a chain memory coupled to the disk I/O engine for buffering a plurality of CPBs to extend to a number of disk transactions scheduled for execution by the disk I/O engine.

20. (Withdrawn) The computer system of claim 19, wherein the bridge component is a Southbridge component.